

Method of transmitting data in networks

Patent Number: DE19712470
Publication date: 1998-10-01
Inventor(s): GROETZNER MICHAEL (DE); KOLATA HOLGER (DE); LAND JUTTA (DE); OCHS GEORG (DE); ROST STEFFEN (DE)
Applicant(s): IBM (US)
Requested Patent: ☐ DE19712470
Application Number: DE19971012470 19970325
Priority Number (s): DE19971012470 19970325
IPC Classification: G06F5/00
EC Classification: G06F17/30B2, H04L29/06
Equivalents:

Abstract

The process is used to transmit data between the program of a transmitting system, e.g. platform A, and the program of a receiving system, platform B, where the program is either installed on a common system or on various systems. The data is structured by a container that defines programming objects and the definition of the structure is given by an application programming interface connecting with the user. The container has stored attributes in various data types, e.g. binary, characteristics, integer and floating point numbers.

Data supplied from the esp@cenet database - I2



⑬ BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

⑩ Off nlegungsschrift
DE 197 12 470 A 1

⑤ Int. Cl.⁶:
G 06 F 5/00

② Aktenzeichen: 197 12 470.4
② Anmeldetag: 25. 3. 97
③ Offenlegungstag: 1. 10. 98

DE 197 12 470 A 1

⑦ Anmelder:
International Business Machines Corp., Armonk,
N.Y., US
⑦ Vertreter:
Teufel, F., Dipl.-Phys., Pat.-Anw., 70569 Stuttgart

⑦ Erfinder:
Ochs, Georg, 71159 Mötzingen, DE; Grätzner,
Michael, 70565 Stuttgart, DE; Land, Jutta, 71093
Weil im Schönbuch, DE; Kolata, Holger, 72108
Rottenburg, DE; Rost, Steffen, 71154 Nufringen, DE

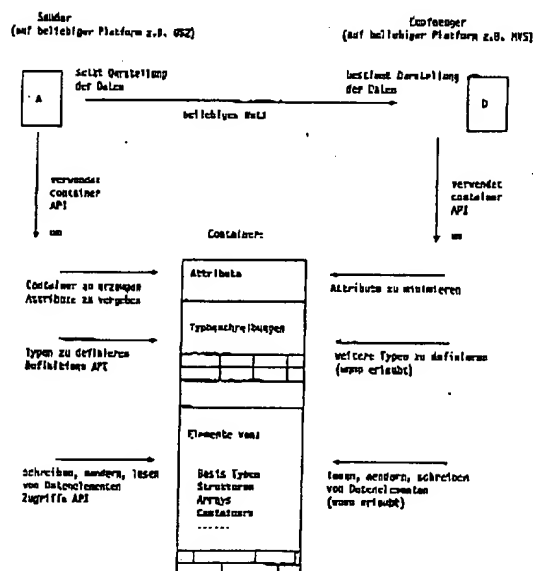
⑤ Entgegenhaltungen:
US 52 78 978

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤ Übertragen von Daten mittels dynamisch programmierbaren Datenobjekten in heterogenen Netzen

⑦ Die vorliegende Erfindung betrifft ein Verfahren zur Übertragung von Daten zwischen verschiedenen Systemen, insbesondere heterogenen Netzen. Die vorliegende Erfindung erlaubt es programmierbare Datentypen zu generieren, in die Daten verschiedener Repräsentation hineingestellt und wieder herausgenommen werden können. Programmierbare Datentypen können sowohl vom Sender als auch vom Empfängersystem generiert werden. Die Struktur der Daten sind im programmierbaren Datentyp selbst niedergelegt und sind frei definierbar. Daher sind Zugriffe auf komplette Strukturen, Arrays als auch einzelne Datenelemente möglich. Der Benutzer erhält die übertragenen Daten in jedem gewünschten Format. Die Daten werden nach Grunddatentypen übertragen, mit der Folge, daß die Übersetzung der Daten einfacher und schneller geht.



DE 197 12 470 A 1

DE 197 12 470 A 1

1

Beschreibung

Die vorliegende Erfindung beschreibt ein Verfahren zum Übertragen von Daten zwischen verschiedenen Systemen, insbesondere eines heterogenen Netzwerks.

Bei der Übertragung von Daten, die unter Vorgaben einer bestimmten Systemhard- bzw. Software erstellt worden sind, stellt sich das Problem der Anpassung der Daten an die Empfänger-Systemhard- bzw. software. Anpassung bedeutet, daß numerische Daten auf die Vorgaben der Datenformate der Empfängerhardware und Charakterdaten auf die vom Empfängersystem unterstützte Software konvertiert werden müssen. Des weiteren muß bei der Übertragung von Charakterdaten Sprachbesonderheiten des Empfängersystems berücksichtigt werden.

Programme, die Daten zwischen verschiedenen Systemen austauschen, verwenden normalerweise strukturierte Daten, die in benutzerdefinierte Datentypen zusammengefaßt sind. Diese Datentypen sind Strukturen, Arrays etc., die untereinander verbunden sein können (Arrays von Strukturen, Strukturen in Strukturen). Zum Zusammenstellen dieser benutzerdefinierten Typen stehen eine Reihe von Grunddatentypen (Ganze Zahlen, Gleitkommazahlen, Charakterdaten etc.) zur Verfügung. Da einige Programme unterschiedliche Datenstrukturen verarbeiten können, muß die Information über die zu übertragende Datenstruktur jederzeit verfügbar sein.

Daten haben immer eine Repräsentation auf den verschiedenen Rechner-Plattformen. Neben der Repräsentation von Ganz- und Gleitkommazahlen, die hardwareabhängig sind, trifft dies verstärkt auf Charakterdaten zu. Zur Darstellung von Buchstaben, Ziffern und Sonderzeichen wird überwiegend der Extended Binary-Coded Interchange Code (EBCDIC) benutzt. Es gibt Rechnerhersteller, die in ihrer Datenverarbeitungsanlage nur den EBCDIC vorgesehen haben und andere Hersteller bei denen sowohl EBCDIC als auch ASCII verwendet werden kann (wie z. B. Unisys).

Hier ist die Abhängigkeit der Repräsentation der Daten sowohl von der Hardware (ASCII \longleftrightarrow EBCDIC) als auch von der Software (Codepage: Deutsch, Französisch, Spanisch, Japanisch) gegeben. Diese Repräsentationen müssen von den Programmen auf den verschiedenen Systemen unterstützt werden.

Mit vielen Datenelemente sind auch Attribute verbunden, die für die Verarbeitung wesentlich sind. Attribute sind z. B. Zugriffsrechte (z. B. was darf der Empfänger mit den Daten machen), Besitz der Daten (wer ist der Besitzer der Daten) oder Anwesenheit von Daten (gibt es das Datenelement und ist es ausgefüllt). Diese Attribute werden normalerweise von dem Absender der Daten vergeben.

Werden zum Beispiel irgendwelche Daten von einer OS/2 Plattform auf eine MVS Plattform übertragen, gibt es derzeit folgende Übertragungsverfahren:

XDR-Verfahren (External Data Repräsentation)

Darum die Daten von der OS/2 Plattform auf die MVS Plattform übertragen werden können, muß sowohl auf der OS/2 Plattform als auch der MVS Plattform eine Beschreibung der zu übertragenden Daten niedergelegt sein. Die Daten werden von der OS/2 Plattform in ihrer vorhandenen Präsentation auf die MVS Plattform übertragen und in die gewünschte Präsentation umgewandelt, die das MVS verarbeiten kann. Dieses Verfahren unterstützt nur eine Darstellung von Charakterdaten, wie z. B. US ASCII \longleftrightarrow US EBCDIC.

Desweiteren unterstützt dieses Verfahren nicht das symbolische Zugreifen auf einzelne Datenelemente aus einer

2

Datenstruktur, z. B. gib mir das Datenelement mit dem Namen "Greetzner" aus den empfangenen Daten.

Auch die Repräsentation der übermittelten Daten kann weder von der Senderseite - hier OS/2 Plattform - noch von der Empfängerseite bestimmt werden. Es kann daher weder vom Anwender festgelegt werden, daß die Daten z. B. in französisch ASCII verschickt werden noch daß die Daten in französisch ASCII beim Empfänger ankommen.

SDDS-Format (Selfdescribing Data Structure)

Dieses Verfahren stellt gegenüber dem XDR-Verfahren eine Erweiterung dahingehend dar, daß auf einzelne Datenelemente einer Datenstruktur zugegriffen werden kann, z. B. gib mir das Datenelement mit dem Namen "Greetzner".

Mit Hilfe dieses Verfahrens kann jedoch nicht auf komplette Datenstrukturen zugegriffen werden, z. B. gib mir den kompletten Datensatz mit dem Namen Greetzner (Name, Kontonummer, Kontostand usw.).

ASN.1-Format (Abstract Syntax Notation)

Die Daten, die übermittelt werden sollen, enthalten als Ergänzung eine Beschreibung der Datenstruktur. Die Daten können daher versendet werden, ohne daß die Empfänger-Plattform die Beschreibung der Datenstruktur kennt.

Voraussetzung für dieses Verfahren ist jedoch, daß Sender/Empfänger den ASN.1 Standard verwenden. ASN.1 ist ein internationaler Standard für Datenaustausch. Die Beschreibung der zu übertragenden Daten erfolgt in einem speziellen von ASN.1 vorgegebenen Format. Dieses Verfahren erlaubt es, sowohl auf einzelne Datenelemente als auch auf komplette Datenstrukturen zuzugreifen.

Bei diesem Verfahren ist jedoch die Repräsentation der Daten weder von der Sender noch der Empfänger-Plattform frei wählbar.

Die Weiterverarbeitung der Daten, z. B. das Aufteilen und Weiterversenden der Daten, erfordert das Zugreifen auf die Beschreibung der Daten.

PCF-Format (Programmable Command Format)

Die Beschreibung der einzelnen Daten wird mitversendet. Auch die Repräsentation wird mitversendet. Dieses Verfahren erlaubt es nicht, auf Strukturen zuzugreifen. Es gibt auch keine Definitionsschnittstelle.

Es ist daher Aufgabe der vorliegenden Erfindung, ein Verfahren zur Übertragung von Daten zu entwickeln, das es erlaubt, Daten innerhalb eines Netzwerkes auf einfache Weise zu übertragen ohne die Einschränkungen der bereits existierenden Verfahren zu haben, wobei die Repräsentation der Daten durch den Sender als auch den Empfänger frei wählbar ist, Datenelemente als auch Datenstrukturen zugreifbar und Datenstrukturen und Attribute frei vom Benutzer definierbar zu machen.

Diese Aufgabe wird durch die Merkmale des Anspruchs 1 gelöst.

Weitere vorteilhafte Ausführungsformen der vorliegenden Erfindung sind in den Unteransprüchen niedergelegt.

Die wesentlichen Vorteile des erfindungsreichen Verfahrens liegen darin, daß die Struktur der Daten in einem Container (programmierbarer Datentyp) selbst gespeichert sind. Dadurch sind keine I/O Zugriffe auf Repository Datenbanken notwendig, um auf einzelne Daten oder Datenelemente zugreifen zu können. Auch sind jederzeit Informationen enthalten, welche Daten gerade im Container vorhanden sind. Dadurch sind auch Inkonsistenzen zwischen Daten und Repository ausgeschlossen.

DE 197 12 470 A 1

3

Auch können die Daten dort zugegriffen werden, wo ein Repository möglicherweise nicht über Netzwerkgrenzen verfügbar gemacht werden kann (Intranet, Internet).

Der Benutzer bekommt seine Daten in jedem gewünschten Format. Dies schließt auch die Wahl der Repräsentation der Daten mit ein.

Die Strukturen der Daten können vom Benutzer frei definiert werden. Anwendungen brauchen sich um die Darstellung der Daten auf anderen Plattformen nicht mehr zu kümmern. Damit entfallen Überprüfungen und Übersetzungstabellenmanagement für diese Anwendungen.

Da die Datenstrukturen selbst im Container (programmierbarer Datentyp) niedergelegt sind, sind Zugriffe auf komplette Strukturen/Arrays als auch Zugriffe auf einzelne Elemente möglich.

Die vorliegende Erfindung wird an Hand eines bevorzugten Ausführungsbeispiels näher erläutert, wobei

Fig. 1 den erfindungsgemäßen Datencontainer (programmierbarer Datentyp) in seinem logischen Format zeigt, in welcher die zu übertragenden Daten auf Senderseite hineingestellt und auf Empfängerseite wieder herausgenommen werden

Fig. 2A die Datenübertragung nach dem Stand der Technik zeigt

Fig. 2B den erfindungsgemäßen Datencontainer in seinem physischen Datenübertragungsformat zeigt, wie die Daten übertragen werden.

Fig. 1 zeigt die Plattform A und B. Auf der Plattform A kann zum Beispiel das Betriebssystem OS/2 und auf der Plattform B das Betriebssystem MVS laufen.

Zwischen der Plattform A und B kann ein beliebiges Netz bestehen. Hierbei wird davon ausgegangen, daß die Daten von der Plattform A gesetzt und die Darstellung der Daten von Plattform B bestimmt werden.

Der Container selbst stellt ein Programmierobjekt dar, in welches Daten hineingestellt und wieder herausgeholt werden können.

Die Struktur der Daten wird im Container selbst gespeichert.

Zur Definition der Struktur der Daten sowie der Daten selbst steht ein API (Application programming Interface) zur Verfügung. Das API ist die Schnittstelle zwischen dem Container und dem Anwendungsentwickler. Neben den vier Grunddatentypen, wie Binär, Charakter, Ganzzahlen und Gleitkommazahlen, sind außerdem Verwaltungsstrukturen, die ihrerseits ein Array, Struktur oder ein Container bilden, vordefiniert. Benutzerdefinierte Datentypen werden durch das Konzept der Strukturbildung mit Hilfe der Grunddatentypen und Verwaltungsstrukturen generiert. Es ist daher möglich, beliebig komplexe Datentypen aus bereits existierenden Datentypen zu definieren.

Der Benutzer bindet mit Hilfe der API Calls die Grunddatentypen mit den Verwaltungsstrukturen und erzeugt somit benutzerdefinierte Datentypen.

Die Definition der Daten für den Container kann dynamisch erfolgen, d. h. neue Daten können zu jedem beliebigen Zeitpunkt dem Container hinzugefügt werden.

Der Empfänger hat die Möglichkeit, neue Datentypen (programmierbare Datentypen) zu definieren und hinzuzufügen. Durch die Definition von Datentypen werden dem Container die Informationen gegeben, damit komplexere Datenstrukturen in den Container hineingestellt und wieder herausgeholt werden können. Sie dienen dazu, die Daten richtig umorganisieren zu können, um sie zu serialisieren bzw. später wieder zu strukturieren. Da jeder Container eine Namensdomäne darstellt, können auf diese Weise gleiche Daten in einem Container gespeichert werden.

Der Zugriff auf die Daten im Container erfolgt über ein

4

Zugriffs APL. Es kann sowohl auf die Typinformation als auch auf die Daten selbst zugegriffen werden. Durch den Zugriff auf die Typinformation ist die Information über das Aussehen der Daten im Container jederzeit verfügbar, d. h. für ein Programm ist es möglich herauszufinden, welche Datenelemente im Container vorhanden sind, welche Werte sie beinhalten und was diese Werte bedeuten. Da die Datenstrukturen im Container bekannt sind, sind Zugriffe unterschiedlicher Granularität möglich.

Im übrigen hat der Benutzer die Möglichkeit, die Repräsentation seiner Daten auf der jeweiligen Plattform durch die Wahl einer Codepage selbst festlegen. Es wird also nicht eine vorgegebene Translationstabelle verwendet, sondern beliebige Übersetzungen sind möglich.

Des weiteren kann durch die Vergabe von Attributen (z. B. Zugriffsrechten) bei der Erzeugung eines Containers der Zugriff auf dessen Daten für einen späteren Empfänger des Containers überprüft werden. Dieser kann den Container z. B. nur zum Lesen öffnen, ohne selbst jedoch neue Daten hinzufügen zu können.

Fig. 2A zeigt wie die Daten herkömmlicherweise übertragen werden.

Die N-Tupel organisierten Daten bestehen (z. B. für den Fall $N = 3$) aus Datentyp (Charakter/Numerisch), Datename und Datenwert. In dieser Reihenfolge werden alle Daten übertragen. Wie sich aus Abb. 2b ergibt, werden nach dem erfindungsgemäßen Verfahren die Daten nach Datentypen übertragen, d. h. zum Beispiel zuerst alle Ganzzahlen, dann alle Charakterdaten, dann alle Fließkommazahlen usw. Dieses Verfahren hat insbesondere den Vorteil, daß die Daten einfach und schnell übersetzt werden. Im Vergleich zum sequentiellen Übersetzen von N-Tupel strukturierten Daten, wo bei N-Datenelementen N Vergleiche gemacht werden müssen, reduziert sich dies bei dem erfindungsgemäßen Verfahren auf die Anzahl der Grunddatentypen.

Patentansprüche

1. Verfahren zur Übertragung von Daten zwischen einem Programm eines Sendersystems und einem Programm eines Empfängersystems, wobei die Programme entweder auf einem gemeinsamen System oder verschiedenen Systemen installiert sind, gekennzeichnet durch folgende Schritte:

- a) Zuordnen der Daten zu deren Beschreibung in Struktur, Repräsentation und Attributen
- b) Übertragen der Daten mit deren Beschreibung in Struktur, Repräsentation und Attributen auf das Empfängersystem
- c) Anpassen der Repräsentation der zu übertragenden Daten an die Vorgabe des Systems.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Daten aus Datenelementen und Datentypen bestehen, und daß jedem Datenelement und jedem Datentyp ein eindeutiger Name zugeordnet ist.

3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß das Anpassen der Repräsentation und der Attribute der Daten durch das Sendersystem erfolgt.

4. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß das Anpassen der Repräsentation und der Attribute der Daten durch das Empfängersystem erfolgt.

5. Verfahren nach Anspruch 3 oder 4, dadurch gekennzeichnet, daß das Anpassen der Daten auf die Vorgaben des Empfängersystems beim Transportieren, beim Empfangen oder nach dem Empfangen erfolgt.

6. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß programmierbare Datentypen generiert werden, die aus Grunddatentypen, Kombination aus

DE 197 12 470 A 1

5

6

Grunddatentypen und programmierbaren Datentypen bestehen können und daß die Daten diesen programmierbaren Datentypen zugeordnet werden.

7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, daß für jeden programmierbaren Datentyp ein Bereich für Namen von Datenelementen, Datenelemente, Datentypen, Repräsentationen und Attributen festgelegt wird.

8. Verfahren nach Anspruch 6 oder 7, dadurch gekennzeichnet, daß die Information Datenname, Datentyp, Repräsentation und Attribut beim Bearbeiten den programmierbaren Datentypen zugeordnet bleiben.

9. Verfahren nach Anspruch 6, 7 oder 8 dadurch gekennzeichnet, daß der programmierbare Datentyp sowohl vom Sender als auch vom Empfänger definiert und abgefragt werden kann.

10. Verfahren nach Anspruch 1 bis 9, dadurch gekennzeichnet, daß die Daten nach Grunddatentypen geordnet zum Empfängersystem übertragen werden.

11. Verfahren nach Anspruch 6 bis 10, dadurch gekennzeichnet, daß die Grunddatentypen, Kombinationen von Grunddatentypen und programmierbare Datentypen bereits vorhanden sind und über Zugriffsfunktionen bekanntgemacht werden.

Hierzu 1 Seite(n) Zeichnungen

25

30

35

40

45

50

55

60

65

- Leerseite -

ZEICHNUNGEN SEITE 1

Nummer:

DE 197 12 470 A1

Int. Cl.⁶:

G 06 F 5/00

Offenlegungstag:

1. Oktober 1998

Sender
(auf beliebiger Plattform z.B. OS2)

Empfänger
(auf beliebiger Plattform z.B. MVS)

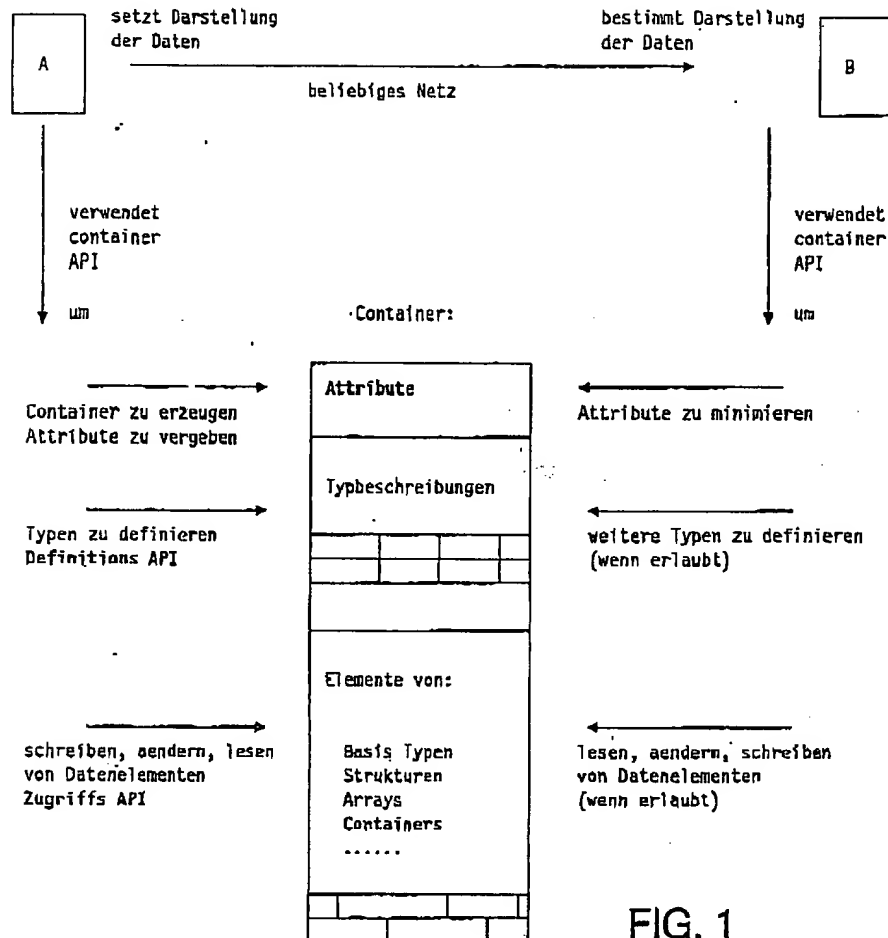


FIG. 1

Typ	Name	Wert	Typ	Name	Wert	...
-----	------	------	-----	------	------	-----

FIG. 2A

Integer Data
Floating Point Data
Binary Data
Character Data
...

FIG. 2B